



**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

**ФАКУЛЬТЕТ**

**«Информатика и системы управления»**

**КАФЕДРА**

**«Программное обеспечение ЭВМ и информационные технологии»**

## **Лабораторная работа № 12**

**Студент:** Керимов А. Ш.

**Группа:** ИУ7-64Б

**Преподаватель:** Толпинская Н. Б.

Москва.  
2020 г.

**Цель работы** — познакомиться со структурой, принципами оформления и логикой выполнения программы на Prolog.

**Задание.** Составить программу – базу знаний, с помощью которой можно определить, например, множество студентов, обучающихся в одном ВУЗе. Студент может одновременно обучаться в нескольких ВУЗах. Привести примеры возможных вариантов вопросов и варианты ответов (не менее 3-х). Описать порядок формирования вариантов ответа.

- Исходную базу знаний сформировать с помощью только фактов.
- Исходную базу знаний сформировать, используя правила.
- Разработать свою базу знаний (содержание произвольно).

## Практическая часть

*Листинг 1. База знаний студентов (факты)*

```
domains
    surname, name, university, faculty, departament, group = string

predicates
    student(surname, name, university, faculty, departament, group)

clauses
    student("Kerimov", "Ahmed", "BMSTU", "IU", "IU7", "IU7-64b").
    student("Novikov", "Mikael", "BMSTU", "IU", "IU7", "IU7-64b").
    student("Yarotskiy", "Danil", "BMSTU", "IU", "IU7", "IU7-61b").
    student("Kerimov", "Ahmed", "MSU", "CMC", "SE", "1").
    student("Ivanov", "Ivan", "MSU", "CMC", "SE", "1").
    student("Ivanov", "Petr", "MSU", "CMC", "SE", "2").
    student("Yugay", "Konstantin", "HSE", "CS", "SE", "BPI-175").
    student("Volkova", "Anna", "DA RFM", "IR", "IR", "1").

goal
    student(Surname, Name, "BMSTU", Faculty, Departament, Group).
    %student("Kerimov", "Ahmed", University, Faculty, Departament, Group).
    %student(Surname, Name, University, Faculty, "SE", Group).
```

```
goal
    student(Surname, Name, "BMSTU", Faculty, Departament, Group).
```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

Surname=Kerimov, Name=Ahmed, Faculty=IU, Departament=IU7, Group=IU7-64b  
Surname=Novikov, Name=Mikael, Faculty=IU, Departament=IU7, Group=IU7-64b  
Surname=Yarotskiy, Name=Danil, Faculty=IU, Departament=IU7, Group=IU7-61b  
3 Solutions

*Рисунок 1 – Студенты из МГТУ*

```
goal
    student("Kerimov", "Ahmed", University, Faculty, Departament, Group).
```

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal\$000.exe]

University=BMSTU, Faculty=IU, Departament=IU7, Group=IU7-64b  
University=MSU, Faculty=CMC, Departament=SE, Group=1  
2 Solutions

*Рисунок 2 – Где учится студент*

```

goal
student(Surname, Name, University, Faculty, "SE", Group).

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
Surname=Kerimov, Name=Ahmed, University=MSU, Faculty=CMC, Group=1
Surname=Ivanov, Name=Ivan, University=MSU, Faculty=CMC, Group=1
Surname=Ivanov, Name=Petr, University=MSU, Faculty=CMC, Group=2
Surname=Yugay, Name=Konstantin, University=HSE, Faculty=CS, Group=BPI-175
4 Solutions

```

Рисунок 3 – Студенты на «Программной инженерии»

Листинг 2. База знаний студентов (факты и правила)

```

domains
    surname, name, university, faculty, departament, group = string

predicates
    student(surname, name, university, faculty, departament, group)
    in_one_group(surname, name, surname, name)
    in_one_university(surname, name, surname, name)

clauses
    student("Kerimov", "Ahmed", "BMSTU", "IU", "IU7", "IU7-64b").
    student("Novikov", "Mikael", "BMSTU", "IU", "IU7", "IU7-64b").
    student("Yarotskiy", "Danil", "BMSTU", "IU", "IU7", "IU7-61b").
    student("Kerimov", "Ahmed", "MSU", "CMC", "SE", "1").
    student("Ivanov", "Ivan", "MSU", "CMC", "SE", "1").
    student("Ivanov", "Petr", "MSU", "CMC", "SE", "2").
    student("Yugay", "Konstantin", "HSE", "CS", "SE", "BPI-175").
    student("Volkova", "Anna", "DA RFM", "IR", "IR", "1").

    in_one_group(Surname1, Name1, Surname2, Name2) :-
        student(Surname1, Name1, University, Faculty, Departament, Group),
        student(Surname2, Name2, University, Faculty, Departament, Group).

    in_one_university(Surname1, Name1, Surname2, Name2) :-
        student(Surname1, Name1, University, _, _, _),
        student(Surname2, Name2, University, _, _, _).

goal
    in_one_group("Kerimov", "Ahmed", "Novikov", "Mikael").
    %in_one_group("Kerimov", "Ahmed", "Yarotskiy", "Danil").
    %in_one_university("Kerimov", "Ahmed", "Yarotskiy", "Danil").

```

```

goal
    in one group("Kerimov", "Ahmed", "Novikov", "Mikael").

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
yes

```

Рисунок 4 - Студенты из одной группы

```

goal
    in one group("Kerimov", "Ahmed", "Yarotskiy", "Danil").

[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
no

```

Рисунок 5 - Студенты не из одной группы

```

goal
in_one_university("Kerimov", "Ahmed", "Yarotskiy", "Danil").
[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
yes|

```

Рисунок 6 - Студенты из одного ВУЗа

Листинг 3. Генеалогическое дерево

```

domains
    name = symbol

predicates
    parent(name, name)
    grandchild(name, name)
    has_child(name)

clauses
    parent(tom, ann).
    parent(tom, mike).
    parent(mary, mike).
    parent(mike, sue).
    parent(mike, john).
    parent(mike, nick).
    parent(john, jane).
    parent(john, jim).

    grandchild(X,Z) :- parent(Y,X), parent(Z,Y).
    has_child(X) :- parent(X,Y).

goal
    grandchild(nick, tom).
    %has_child(john).
    %has_child(jane).

```

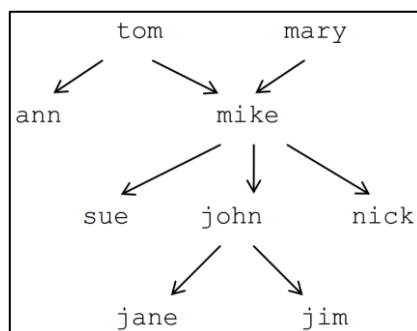


Рисунок 7 - Генеалогическое дерево

```

goal
grandchild(nick, tom).
[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
yes

```

Рисунок 8 – Ник — правнук Тома

```
goal
has_child(john).
[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
yes
```

Рисунок 9 - У Джона есть потомство

```
goal
has_child(jane).
[Inactive C:\VIP52\BIN\WIN\32\Obj\goal$000.exe]
no
```

Рисунок 10 - У Джейн нет потомства

## Теоретическая часть

Программа на Prolog представляет собой **базу знаний** и **вопрос**.

**База знаний** состоит из предложений — **CLAUSES** (отдельных знаний или утверждений): **фактов** и **правил**. Каждое предложение заканчивается точкой. Предложения бывают двух видов: факты и правила.

Предложение более общего вида — **правило** — имеет вид:

$A :- B_1, \dots, B_n.$

**A** называется **заголовком правила**, а **B<sub>1</sub>, ..., B<sub>n</sub>** — **телом правила**.

**Факт** — это частный случай правила. Факт — это предложение, в котором отсутствует тело (т. е. тело пустое).

Третьим специфическим видом предложений Prolog можно считать **вопросы**. Вопрос состоит только из тела – составного терма (или нескольких составных термов). Вопросы используются для выяснения выполнимости некоторого отношения между описанными в программе объектами. Система рассматривает вопрос как цель, к которой (к истинности которой) надо стремиться. Ответ на вопрос может оказаться логически положительным или отрицательным, в зависимости от того, может ли быть достигнута соответствующая цель.

Программа на Prolog может содержать вопрос в программе (так называемая **внутренняя цель GOAL**). Если программа содержит внутреннюю цель, то после запуска программы на выполнение система проверяет достижимость заданной цели, исходя из базы знаний.

Ответ на поставленный вопрос система дает в логической форме — «Да» или «нет». Цель системы состоит в том, чтобы на поставленный вопрос найти возможность, исходя из базы знаний, ответить «Да». Вариантов ответить «Да» на поставленный вопрос может быть несколько.

Основным элементом языка является терм. Терм — это:

1. **Константа:**

- Число (целое, вещественное),
- Символьный атом (aA, A\_2),
- Строка: последовательность символов, заключенных в кавычки,

2. **Переменная:**

- Именованная — обозначается комбинацией символов латинского алфавита, цифр и символа подчеркивания, начинающейся с прописной буквы или символа подчеркивания (X, A21, \_X),
- Анонимная — обозначается символом подчеркивания (  ),

### 3. Составной терм:

- $f(t_1, t_2, \dots, t_m)$ , где  $f$ -функтор (функциональный символ),  $t_1, t_2, \dots, t_m$  — термы, в том числе и составные.

## Особенность использования переменных

Именованные переменные уникальны в предикатах одного предложения, анонимные уникальны везде. Анонимные переменные не возвращают значение. Переменной можно обозначить любой объект. При описании переменная может потерять свое значение, но потом его можно вернуть.

### Структура программы

Программа состоит из разделов

- a. Domains – описание имен и структур объектов (не обязателен)
- b. Predicates – описание предикатов (названий отношений между объектами)
- c. Clauses – база знаний
- d. Goal – Раздел целевых утверждений

Программа состоит из предложений

- Факт (безусловная истина, формулируется составным термом)- частный случай правил.
- Правила (условная истина, способ порождения новых фактов на основе имеющихся)

○ Пример:

A:- B1, B2,B3. (правило)

A – заголовок, B1, B2,B3.- тело

Вопрос:

- Конъюнктивный (B1, B2, B3)
- Дизъюнктивный (B1; B2; B3).